Robust Path Planning and Feedback Design under Stochastic Uncertainty

Lars Blackmore*

Autonomous vehicles require optimal path planning algorithms to achieve mission goals while avoiding obstacles and being robust to uncertainties. The uncertainties arise from exogenous disturbances, modeling errors, and sensor noise, which can be characterized via stochastic models. Previous work defined a notion of robustness in a stochastic setting by using the concept of chance constraints. This requires that mission constraint violation can occur with a probability less than a prescribed value.

In this paper we describe a novel method for optimal chance constrained path planning with feedback design. The approach optimizes both the reference trajectory to be followed and the feedback controller used to reject uncertainty. Our method extends recent results in constrained control synthesis based on convex optimization to solve control problems with nonconvex constraints. This extension is essential for path planning problems, which inherently have nonconvex obstacle avoidance constraints. Unlike previous approaches to chance constrained path planning, the new approach optimizes the feedback gain as well as the reference trajectory.

The key idea is to couple a fast, nonconvex solver that does not take into account uncertainty, with existing robust approaches that apply only to convex feasible regions. By alternating between robust and nonrobust solutions, the new algorithm guarantees convergence to a global optimum. We apply the new method to an unmanned aircraft and show simulation results that demonstrate the efficacy of the approach.

I. Introduction

Autonomous vehicles such as Unmanned Air Vehicles (UAVs) need to be able to plan trajectories to a specified goal that avoid obstacles, and are robust to the uncertainty that arises in the real world. Sources of uncertainty include uncertain state estimation, disturbances and modeling errors. While much prior research has focused on robustness to set-bounded uncertainty,[1–4] many sources of uncertainty, such as wind disturbances, are most naturally characterized using stochastic models.[5] With stochastic uncertainty, it is typically not possible to guarantee mission success, defined as reaching the goal region and avoiding all obstacles, since there is always a small probability that a very large disturbance will occur. We can, however, define robustness in terms of *chance constraints*. These require that mission failure occurs with at most a user-specified probability. Such constraints enable the operator to trade conservatism against performance; a plan with a very low probability of failure will typically require more fuel, or time, to complete. In this paper we are concerned with the problem of optimal chance constrained path planning with feedback design. That is, we would like to design a sequence of feedforward control inputs and a feedback controller that minimizes cost, such as fuel use, while ensuring that the probability of failure is below the required threshold. We are concerned with discrete-time linear systems; prior work showed that a UAV operating at a constant altitude as well as other autonomous vehicles can be approximated as such a system, subject to velocity and turn rate constraints.[6,7] A number of recent articles have addressed parts of this problem, which we summarize here.

In the path planning (or obstacle avoidance) problem, the feasible region for the system state is almost always nonconvex. Recent work has addressed the problem of chance constrained planning and control in nonconvex feasible regions.[8–10] This work noted that in stochastic systems, the use of a feedback controller to reject disturbances is essential; otherwise the uncertainty in the predicted state grows in time without

*Guidance and Control Analysis Group, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA.

bound. Previous work therefore assumed a feedback controller as well as a feedforward control (or *reference trajectory*). However the feedback controller had to be specified ahead of time by hand, rather than being incorporated into the optimization explicitly. In this paper, we aim to find the optimal feedback controller as well as the optimal feedforward control.

Feedback design was considered explicitly in the case of control within convex feasible regions by a series of recent articles.[11–15] The approach of Ref. 15 converts chance constraints into set constraints on the state mean. The problem of optimal feedforward and control law design is then posed as a Second Order Cone Program (SOCP) and solved using efficient existing techniques.[16] The tractability of this approach relies on the convexity of the feasible region; without this, the resulting optimization is nonlinear and nonconvex. This means that finding the global optimum is intractable in the general case.

In this paper we extend the work of Ref. 15 to nonconvex feasible regions, that is, to the problem of robust path planning. The key idea is to couple a fast, nonconvex solver[6] that does not take into account uncertainty, with the convex robust approach of Ref. 15. The intuition here is that, in many cases, the optimal robust solution is close to the nonrobust solution. We therefore use the fast nonrobust solver to identify convex regions in which the robust solver should search for a solution to the robust path planning problem. By alternating between robust and nonrobust solutions, the new algorithm guarantees convergence to a global optimum, while using bounding arguments to ensure that a small subset of the possible convex regions are explored.

II. Problem Statement

In this paper, we are concerned with the following discrete-time linear plant:

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + B^w\mathbf{w}_k + B\mathbf{u}_k \\ \mathbf{y}_k &= C\mathbf{x}_k + D\mathbf{w}_k. \end{aligned} \tag{1}$$

Here $\mathbf{x} \in \Re^{n_x}$ is the system state, $\mathbf{y} \in \Re^{n_y}$ are the observable system outputs, $\mathbf{u} \in \Re^{n_u}$ are the system inputs, and $\mathbf{w}$ is a noise vector. The noise vector can model disturbances, uncertainty in the system model, and sensor noise. We assume that $\mathbf{w}$ is a Gaussian noise process and that the initial state $\mathbf{x}_0$ is a Gaussian random variable; these two are uncorrelated. We use $\mathbf{x}_k$ to denote the value of $\mathbf{x}$ at time step k, and $\mathbf{x}'$ to denote the transpose of $\mathbf{x}$. We use $P(A)$ to denote the probability of event A and $p(\mathbf{x})$ to denote the probability distribution function of random variable $\mathbf{x}$. We use $\bar{\mathbf{x}}$ to denote the mean of the random variable $\mathbf{x}$.

We refer to the plant in the absence of uncertainty as the *reference plant*, defined as:

$$\begin{aligned} \mathbf{x}^r_{k+1} &= A\mathbf{x}^r_k + B^w\bar{\mathbf{w}}_k + B\mathbf{u}^r_k \\ \mathbf{y}^r_k &= C\mathbf{x}^r_k + D\bar{\mathbf{w}}_k. \end{aligned} \tag{2}$$

Note that here, the noise variables have been set to their mean values. We assume that a feedback controller is used to reject disturbances and drive the system state to the reference value $\mathbf{x}^r$. In the path planning problem we aim to design both a feedback map and the feedforward (or *reference*) control input $\mathbf{u}^r$. We assume a time-varying linear feedback map such that:

$$\mathbf{u}_k = \mathbf{u}^r_k + \sum_{t=0}^{k} K_{kt}(\mathbf{y}_t - \mathbf{y}^r_t). \tag{3}$$

In other words, K_{kt} is the feedback gain that, at time k, multiplies the error between the actual observation and the reference observation at time t. This controller is non-anticipating, i.e. the control input at time k does not depend on future observations. A special case of the controller in (3) is the fixed linear gain feedback $\mathbf{u}_k = \mathbf{u}^r_k + K(\mathbf{y}_k - \mathbf{y}^r_k)$. The feedback structure is illustrated in Figure 1.

In the path planning problem, we plan over a finite horizon of time instances from $k = 0$ to $k = T$. For notational convenience we 'lift' the variables of interest over the time horizon using the following definitions:

$$\mathbb{X} = \begin{bmatrix}\mathbf{x}_0\\ \mathbf{x}_1\\ \vdots\\ \mathbf{x}_T\end{bmatrix} \quad \mathbb{X}^r = \begin{bmatrix}\mathbf{x}^r_0\\ \mathbf{x}^r_1\\ \vdots\\ \mathbf{x}^r_T\end{bmatrix} \quad \mathbb{Y} = \begin{bmatrix}\mathbf{y}_0\\ \mathbf{y}_1\\ \vdots\\ \mathbf{y}_T\end{bmatrix} \quad \mathbb{Y}^r = \begin{bmatrix}\mathbf{y}^r_0\\ \mathbf{y}^r_1\\ \vdots\\ \mathbf{y}^r_T\end{bmatrix} \quad \mathbb{U} = \begin{bmatrix}\mathbf{u}_0\\ \mathbf{u}_1\\ \vdots\\ \mathbf{u}_T\end{bmatrix} \quad \mathbb{U}^r = \begin{bmatrix}\mathbf{u}^r_0\\ \mathbf{u}^r_1\\ \vdots\\ \mathbf{u}^r_T\end{bmatrix} \quad \mathbb{W} = \begin{bmatrix}\mathbf{w}_0\\ \mathbf{w}_1\\ \vdots\\ \mathbf{w}_T.\end{bmatrix} \tag{4}$$

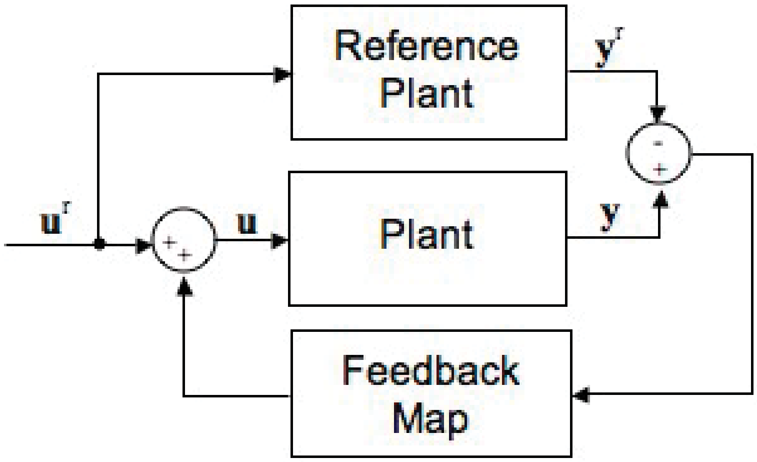

Figure 1. Control structure for robust path planning problem with stochastic plant. The feedforward control is used to set the reference state value, while feedback control is used to drive the system state to the reference value in the presence of noise.

The initial state mean and covariance are denoted $\bar{\mathbf{x}}_0$ and P_0 respectively. The mean and covariance of the noise sequence $\mathbb{W}$ are denoted $\bar{\mathbb{W}}$ and V respectively. The open-loop lifted system dynamics are given by:

$$\begin{aligned} \mathbb{X} &= G_{xx}\mathbf{x}_0 + G_{xu}\mathbb{U} + G_{xw}\mathbb{W} \\ \mathbb{Y} &= G_{yx}\mathbf{x}_0 + G_{yu}\mathbb{U} + G_{yw}\mathbb{W}, \end{aligned} \tag{5}$$

and the open loop lifted reference dynamics are given by:

$$\begin{aligned} \mathbb{X}^r &= G_{xx}\bar{\mathbf{x}}_0 + G_{xu}\mathbb{U}^r + G_{xw}\bar{\mathbb{W}} \\ \mathbb{Y}^r &= G_{yx}\bar{\mathbf{x}}_0 + G_{yu}\mathbb{U}^r + G_{yw}\bar{\mathbb{W}}, \end{aligned} \tag{6}$$

where the matrices G_{xx}, G_{xu}, G_{xw}, G_{yx}, G_{yu} and G_{yw} are calculated through repeated application of the system definition (1). Note that, since the dynamics equations (5) are linear, we have $E[\mathbb{X}] = \mathbb{X}^r$ and $E[\mathbb{Y}] = \mathbb{Y}^r$. The control structure can be expressed as $\mathbb{U} = \mathbb{U}^r + K(\mathbb{Y} - \mathbb{Y}^r)$, where:

$$K = \begin{bmatrix} K_{00} & O & \cdots & O \\ K_{10} & K_{11} & \cdots & O \\ \vdots & \vdots & & \vdots \\ K_{N0} & K_{N1} & \cdots & K_{NN} \end{bmatrix}. \tag{7}$$

The lower block triangular structure of K ensures that the controller is non-anticipating. We denote the set of all possible lower block triangular matrices of the form (7) as $\mathcal{K}$.

We define a feasible region $F_x \subset \Re^{n_x \cdot (T+1)}$ in the space of the lifted state $\mathbb{X}$, and a feasible region $F_u \subset \Re^{n_u \cdot (T+1)}$ in the space of the lifted control inputs $\mathbb{U}$, both of which *may be nonconvex.* We also define a cost $h(\mathbb{U}^r, \mathbb{X}^r)$, and assume that $h(\cdot)$ is a piecewise linear function. The path planning problem may now be stated as:

Definition 1. *The* chance constrained path planning problem *consists of solving the following optimization problem:*

$$\begin{aligned} &\textit{Minimize } h(\mathbb{U}^r, \mathbb{X}^r) \textit{ over } \mathbb{U}^r \textit{ and } K \\ &\textit{subject to} \\ &\mathbb{U}^r \in F_u \\ &P(\mathbb{X} \notin F_x) \le \delta \\ &(3), (5), (6), (7) \end{aligned}$$

III. Summary of existing work

In Section IV we describe a new algorithm for solving the chance constrained path planning problem (Def. 1). The new algorithm builds from two prior algorithms that solve special cases of the problem in Def. 1. The first uses Mixed Integer Linear Programming (MILP) to perform path planning for deterministic linear systems. While this approach can deal with nonconvex feasible regions, it *does not take into account uncertainty.* The second uses convex optimization to solve chance constrained feedback control problems for stochastic linear systems *in convex regions.* In the following sections we describe the key properties of these two approaches as they relate to the new nonconvex robust path planning algorithm.

A. Nonrobust Path Planning using MILP

Definition 2. *The* nonrobust path planning problem *consists of solving the following optimization problem:*

Minimize $h(\mathbb{U}^r, \mathbb{X}^r)$ *over* $\mathbb{U}^r$
subject to
$\mathbb{U}^r \in F_u$
$\mathbb{X}^r \in F_x$
(6)

Compared with the robust path planning problem (Def. 1), the key difference in Def. 2 is that we are only concerned with the reference state trajectory. The reference state trajectory is deterministic, that is, it does not model any sources of uncertainty. This means that in Def. 2, we constrain the reference system state to remain within the feasible region with certainty. In addition, we no longer design a feedback law, since feedback is only necessary to drive the actual state to the reference state in the presence of uncertainty.

Ref. 6 shows that, in the case of polygonal feasible regions, the nonrobust path planning problem can be posed as a Mixed Integer Linear Program. Efficient commercially-available software[17] enables fast solution of the resulting MILP, and guarantees that the globally optimal solution can be found in bounded time. The key idea behind the approach of Ref. 6 is to encode obstacle avoidance constraints as disjunctions of linear constraints. These disjunctions can be expressed in a MILP formulation using binary variables and 'Big-M' techniques as follows.

A polygonal convex feasible region can be expressed as a conjunction of linear constraints, as illustrated in Figure 2. The system state at time step k is in the feasible region if and only if:

$$\bigwedge_{l} \mathbf{a}'_{kl}\mathbf{x}^r_k \leq b_{kl} \tag{8}$$

Using such constraints we can ensure that the system state ends in a defined goal region, and remains in a convex safe region at all time steps. An arbitrary polygonal non-convex feasible region can be described by removing convex infeasible regions, or obstacles, from the safe region, as illustrated in Figure 2. The reference trajectory avoids a given obstacle O_j, illustrated in Figure 2, if and only if:

$$\bigwedge_{k} \bigvee_{l} \mathbf{a}'_{jl}\mathbf{x}^r_k \geq b_{jl} \tag{9}$$

Then the state is in the nonconvex feasible region if and only if:

$$\Big(\bigwedge_{j} \bigwedge_{k} \bigvee_{l} \mathbf{a}'_{jl}\mathbf{x}^r_k \geq b_{jl}\Big) \bigwedge \Big(\bigwedge_{k} \bigwedge_{l} \mathbf{a}'_{kl}\mathbf{x}^r_k \leq b_{kl}\Big) \tag{10}$$

The challenge is to encode the disjunctions in (9). As shown by Ref. 6, in order to encode avoidance of obstacle O_j, we introduce binary variables $z(j,k,l) \in \{0,1\}$ that indicate whether a given constraint l for a given obstacle O_j is satisfied at a given time step k. The constraint:

$$\mathbf{a}'_{jl}\mathbf{x}^r_t - \mathbf{b}_{jl} + Mz(j,k,l) \geq 0, \tag{11}$$

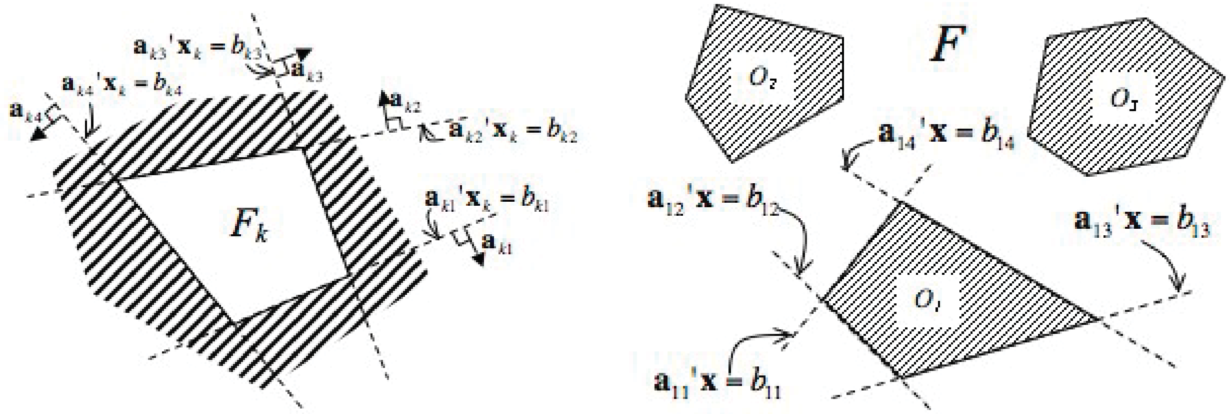

Figure 2. Left: Polygonal convex feasible region F_k for the state at time step k encoded using a conjunction of linear constaints. The state is in the feasible region if all of the linear constraints are satisfied. Right: Two-dimensional non-convex polygonal feasible region F. The feasible region is the complement of several convex obstacles (shaded). Each obstacle O_j is defined by the N_j vector normals $\mathbf{a}_{j1}, \ldots, \mathbf{a}_{jN_j}$.

means that $z(j,k,l) = 0$ implies that constraint l in obstacle O_j is satisfied at time step k. Here M is a large positive constant. We can now encode constraint (9) in terms of the binary variables as follows:

$$\sum_{l=1}^{N_j} z(j,k,l) \leq N_j - 1 \quad \forall k. \tag{12}$$

By imposing the constraint (12) for all obstacles, we ensure that all obstacles are avoided at all time steps. Using further binary variables we can encode nonconvex constraints on the reference control input sequence $\mathbb{U}^r$. Hence the constraints $\mathbb{U}^r \in F_u$ and $\mathbb{X}^r \in F_x$ are encoded as linear constraints involving binary variables. The dynamics constraints (6) are linear, and the cost function $h(\mathbb{U}^r, \mathbb{X}^r)$ is linear. Hence the nonrobust path planning problem (Def. 2) is a MILP, and can be solved to global optimality.

We denote the set of all binary variables $z(\cdot)$ as $\mathbb{Z}$. Note that each full assignment to $\mathbb{Z}$ corresponds to a convex polygonal region that is a subset of the full nonconvex feasible space. The MILP solution of the nonrobust path planning problem therefore returns not only the optimal control sequence $\mathbb{U}^r$, but also the convex region of F_x in which the optimal reference trajectory $\mathbb{X}^r$ was found. We use this property in our new algorithm for robust path planning, described in Section IV. For notational convenience in this paper, we define the function that solves the nonrobust path planning problem using MILP in Table 1.

	Function NonRobustPathPlanning(F_x, F_u) **returns** $h, \hat{\mathbb{U}}^r, \mathbb{Z}$
1)	Express the polygonal nonconvex region F_x in a Mixed Integer Linear format using (11) and (12).
2)	Solve the optimization problem in Def. 2 for $\hat{\mathbb{U}}^r$ and $\mathbb{Z}$ using MILP techniques, as described by Ref. 6.

Table 1. Nonrobust Path Planning using MILP.

B. Robust Control Design using Convex Optimization

Definition 3. *The* chance constrained convex planning problem *consists of solving the following optimization problem:*

$$\begin{aligned}
&\textit{Minimize } h(\mathbb{U}^r, \mathbb{X}^r) \textit{ over } \mathbb{U}^r \textit{ and } K \\
&\textit{subject to} \\
&\mathbb{U}^r \in F_u \\
&P(\mathbb{X} \notin F_x) \leq \delta \\
&F_u, F_x \textit{ convex} \\
&(3), (5), (6), (7).
\end{aligned} \tag{13}$$

This is identical to the chance constrained path planning problem (Def. 1) except that the feasible sets F_u and F_x are restricted to being convex. This problem was addressed in a series of recent articles[11–14] and expanded upon in Ref. 15. This work determines the distance of the reference state $\mathbb{X}^r$ from the boundaries of F_x sufficient to ensure that the chance constraint $P(\mathbb{X} \notin F_x) \leq \delta$ is satisfied for any $\delta \leq 0.5$. This enables the chance constraint $P(\mathbb{X} \notin F_x) \leq \delta$ to be approximated as a constraint on the $\mathbb{X}^r$, which is not a random variable. The problem in Def. 3 is then approximated as follows:

Definition 4. *The* conservative convex planning problem *consists of solving the following optimization problem:*

$$\begin{aligned}
&\textit{Minimize } h(\mathbb{U}^r, \mathbb{X}^r) \textit{ over } \mathbb{U}^r \textit{ and } K \\
&\textit{subject to} \\
&\mathbb{U}^r \in F_u \\
&\mathbb{X}^r + \mathcal{E}(K, \delta) \subset F_x \\
&F_u, F_x \textit{ convex} \\
&(3), (5), (6), (7),
\end{aligned} \tag{14}$$

where $\mathcal{E}(r, K, \delta)$ is an ellipsoid defined such that:

$$P\big(\mathbb{X} \notin \{\mathbb{X}^r + \mathcal{E}(r, K, \delta)\}\big) \leq \delta. \tag{15}$$

From (15) and Def. 4 we see that satisfaction of the constraint $\mathbb{X}^r + \mathcal{E}(r, K, \delta) \subset F_x$ implies satisfaction of the chance constraint $P(\mathbb{X} \notin F_x) \leq \delta$. Hence a feasible solution to the conservative chance constrained convex planning problem (Def. 4) is a feasible solution to the chance constrained convex planning problem (Def. 3). Ref. 15 shows that the optimization in Def. 4 can be posed as a Second-Order Cone Program (SOCP). The details of this are shown in the Appendix. An SOCP is an example of a convex optimization problem, for which fast algorithms exist with guaranteed convergence to global optimality, with known bounds on the convergence rate.[16] Hence the approximation of the chance constrained convex planning problem can be solved efficiently. However the ellipsoidal approximation of the chance constraint in Def. 4 introduces conservatism; this introduces suboptimality in the returned solution. In the general case there exist non-ellipsoidal sets that satisfy (15) but that yield a lower cost solution to the Def. 4; however the ellipsoid constraint is required to pose the problem as an SOCP. For notational simplicity, we define the function that solves the conservative convex planning problem using MILP in Table 2.

	Function CONSERVATIVECONVEXPLANNING(F_x, F_u, δ) **returns** $h, \mathbb{U}^r, K$
1)	Express the conservative convex planning problem in Def. 4 as a SOCP using the approach in the Appendix.
2)	Solve the SOCP using existing techniques[16] for the optimal solution $\{\mathbb{U}^r, K\}$ with cost h.

Table 2. Nonrobust Path Planning using MILP.

IV. A New Algorithm for Robust Path Planning

In this paper we extend the work of Ref. 15 to the case of nonconvex feasible regions, in other words to path planning with obstacles. We use the results of Ref. 15 to approximate the chance constraints in a conservative manner. This results in the following approximation of the chance constrained path planning problem:

Definition 5. *The* conservative path planning problem *consists of solving the following optimization problem:*

$$\begin{aligned}
&\textit{Minimize } h(\mathbb{U}^r, \mathbb{X}^r) \textit{ over } \mathbb{U}^r \textit{ and } K \\
&\textit{subject to} \\
&\mathbb{U}^r \in F_u \\
&\mathbb{X}^r + \mathcal{E}(r, K, \delta) \subset F_x \\
&(3), (5), (6), (7), (15).
\end{aligned} \tag{16}$$

From the result of Ref. 15, any feasible solution to the conservative path planning problem (Def. 5) is guaranteed to be a feasible solution to the chance constrained path planning problem (Def. 4). Note that the only difference between Def. 5 and Def. 4 is that the feasible regions are no longer required to be convex. The extension of the algorithm proposed by Ref. 15 to the nonconvex case, however, is far from trivial. The tractability of the conservative convex planning problem (Def. 4) is crucially dependent on the convexity of the sets F_u and F_x. If either of these are nonconvex, the resulting optimization is nonlinear and nonconvex. Finding the globally optimal solution of such a problem is intractable in the general case; existing algorithms provide, in practice, convergence to local optima and hence require good initial guesses for acceptable performance.

An alternative approach would be to pose the conservative path planning problem as a Mixed Integer Convex Program (MICP). As in Section III-A, nonconvex polygonal constraints can be encoded using binary variables. For a given assignment to the binary variables, the constraints are convex, and we therefore have an MICP. Recent development in solver technology has enabled the solution of such problems, using convex optimizers to solve convex subproblems, and branch-and-bound to search the nonconvex space for the globally optimal solution efficiently. For path planning problems of interest, however, the resulting MICPs are too large to be tractable. We therefore require a new, tractable algorithm for solving the conservative path planning problem. In this section we describe such an algorithm.

A. Algorithm Description

The key idea behind the new algorithm is to use the approaches of Ref. 6 and Ref. 15 to solve subproblems of the chance constrained path planning problem in such a manner that we achieve good average-case performance while guaranteeing convergence to a global optimum in finite time. Since the nonrobust path planning problem (Def. 2) is a MILP, it can be solved extremely quickly. The conservative convex planning problem (Def. 4) is significantly slower to solve. Intuitively, however, the optimal robust solution is close to the optimal nonrobust solution in many cases. In addition, the optimal nonrobust cost is a lower bound on the optimal robust cost, as we prove in Section IV-B. We therefore use the nonrobust solution to guide the search for a robust solution in two ways; first, to identify promising regions in which to search for a robust solution; second, to terminate the search by certifying that the robust solution cannot improve.

The algorithm proceeds as follows. First we solve the nonrobust problem and identify the convex region in which the optimal solution was found. We then look for a robust solution in this convex feasible region using the conservative convex planning approach of Ref. 15, described in Section III-B. Using the results of the robust optimization we remove regions of the nonconvex feasible region in which we know the robust cost cannot improve. The algorithm then starts another iteration by solving the nonrobust problem in the diminished nonconvex feasible space. This proceeds until the optimal nonrobust cost is greater than the best robust cost found so far. Since the optimal nonrobust cost is a lower bound on the robust cost in the remaining feasible space, we are guaranteed at this point to have found the globally optimal robust solution. As we demonstrate empirically in Section V, the algorithm finds the globally optimal solution quickly, and then spends the rest of its running time proving that the solution found is indeed globally optimal. Pseudocode for the algorithm is given in Table 3. In Section B we prove that the algorithm reaches a global optimum in finite time.

B. Algorithm Properties

Lemma 1 (Conservative Cost Greater than Nonrobust Cost). *Fix all problem parameters including the feasible regions F_x and F_u. Denote the cost of the optimal solution to the nonrobust path planning problem (Def. 2) as $\hat{h}$. Denote the cost of the optimal solution to the conservative path planning problem (Def. 5) as h^*. If no feasible solution exists, the optimal cost is defined as infinity. Then, for any $\delta \leq 0.5$, we have $\hat{h} \leq h^*$.*

Proof: In the nonrobust path planning problem, the state constraints ensure that the reference state is inside the feasible region, i.e. $\mathbb{X}^r \in F_x$. The conservative path planning problem with $\delta \leq 0.5$ ensures that the reference state is at least a certain backoff distance $r > 0$ from the boundaries of F_x. Hence the state constraints are strictly tighter in the conservative path planning problem. Other than this, the two problems are identical. Hence the conservative cost cannot be less than the nonrobust cost; equality occurs when the state constraints are not tight in the optimal solution. □

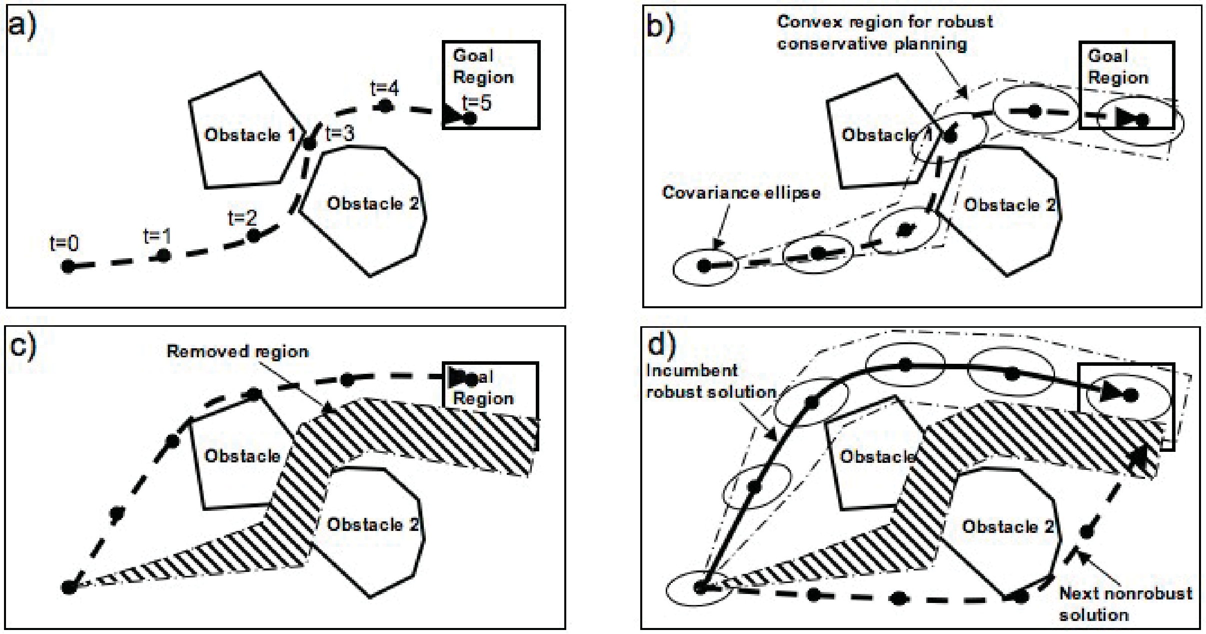

Figure 3. Illustration of new approach to chance constrained path planning. a) First, a nonrobust optimal solution is found, which does not take into account uncertainty. The algorithm identifies the convex region $\mathcal{C}$ in which this solution lies. b) The algorithm searches for a chance constrained (robust) solution in $\mathcal{C}$ (note that a region convex in the lifted vector $\mathbb{X}$ appears non-convex in the figure). No robust solution exists in the $\mathcal{C}$. We remove from the search space a region guaranteed not to contain a feasible robust solution. c) We search for a nonrobust optimal solution, which this times avoids the narrow corridor. d) A robust feasible solution does exist in the convex region about the solution in c). This becomes our new incumbent solution. Our next nonrobust solution has cost greater than our incumbent cost. Since this is a lower bound on the robust cost in the remaining search space, we have guaranteed optimality.

Lemma 1 says that the conservative cost is never less than the nonrobust cost for a given set of problem parameters. Following directly from this, we have $\hat{h} = \infty \implies h^* = \infty$ and $h^* < \infty \implies \hat{h} < \infty$. In other words, if the nonrobust problem is infeasible, then the conservative problem is infeasible, and if the conservative problem is feasible, then the nonrobust problem is feasible.

Lemma 2 (Pruning). *At iteration i, the function* PRUNESEARCHSPACE *does not remove any feasible solution with cost better than $\tilde{h}_i$.*

Proof: The function PRUNESEARCHSPACE identifies which of the obstacle constraints are tight in the optimal conservative solution, which has cost $\tilde{h}_i$. We denote this subset *tight_constraints*. PRUNESEARCHSPACE removes the part of the nonconvex search space for which *tight_constraints* are imposed. Any conservative path planning problem with *tight_constraints* imposed will have an optimal cost no better than $\tilde{h}_i$. Hence any feasible solution removed by PRUNESEARCHSPACE has cost no better than $\tilde{h}_i$. □

Theorem 1 (Global Optimality). *The algorithm described in Table 3 terminates only if the globally optimal feasible solution has been found, or no feasible solution exists.*

Proof: The algorithm starts with a nonconvex feasible state region F_x. At each iteration i, in Step 9, a subset P_i of the feasible region is removed, leaving a region F'_x. From Lemma 2 we know that for each i, $\tilde{h}_i$ is the minimum cost in P_i. We set h^* to $\tilde{h}_i$ if and only if $\tilde{h}_i < h^*$, so $h^* = \min_i \tilde{h}_i$. Hence at Step 9, we know that h^* is the minimum cost feasible solution in the set $R := \bigcup_{j=1}^{i} P_j$. That is, our incumbent cost is guaranteed to be no worse than any feasible solution in the search space explored so far. The algorithm terminates at iteration j only if $\hat{h}_j \geq h^*$. The nonrobust cost $\hat{h}_j$ is the optimal nonrobust solution in the remaining (possibly nonconvex) feasible region F'_x. From Lemma 1 we know that the optimal conservative cost in F'_x is no less than $\hat{h}_j$. Hence any conservative feasible solution in F'_x has no better cost than h^*. Since we know h^* is the minimum of all conservative feasible costs in R, and that $R \cup F'_x = F_x$, at termination h^* is the globally optimal cost in F_x. If the globally optimal cost is infinite, then no conservative feasible solution exists. □

	Function RobustNonconvexMain(F_x, F_u, δ) **returns** $\mathbb{U}^r$,K,$\bar{h}$,h^*
1)	Initialize $\bar{h} \leftarrow +\infty$, $h^* \leftarrow +\infty$, $i \leftarrow \emptyset$, $globaloptimal \leftarrow false$, $F_x' \leftarrow F_x$.
2)	Increment i
3)	Solve nonrobust problem (Def. 2) on feasible state region F_x': $\{\hat{h}_i, \hat{\mathbb{U}}^r, \mathbb{Z}\} \leftarrow$NonrobustPathPlanning$(F_x', F_u)$
4)	if $\hat{h}_i \geq h^*$ then $globaloptimal \leftarrow true$.
5)	if $\hat{h}_i < \bar{h}$ then $\bar{h} \leftarrow \hat{h}$.
6)	Form convex subregion: $\tilde{F}_x \leftarrow$FormConvexRegion$(F_x, \mathbb{Z})$
7)	Solve convex conservative problem: $\{\tilde{h}_i, \tilde{\mathbb{U}}^r, \tilde{K}\} \leftarrow$ConservativeConvexPlanning$(\tilde{F}_x, F_u, \delta)$
8)	Check for a new incumbent: If $\tilde{h}_i < h^*$ then $h^* \leftarrow \tilde{h}_i$, $\mathbb{U}^r \leftarrow \tilde{\mathbb{U}}^r$, $K \leftarrow \tilde{K}$
9)	Prune search space: $F_x' \leftarrow$PruneSearchSpace$(\tilde{\mathbb{U}}^r, \tilde{K}, \tilde{F}_x, F_x', \mathbb{Z})$
10)	if ($globaloptimal$ = false) go to Step 2) else terminate

Table 3. Algorithm for Robust Path Planning.

	Function FormConvexRegion$(F_x, \mathbb{Z})$ **returns** $\tilde{F}_x$
1)	Express nonconvex region F_x in Mixed Integer Linear format as described in Section A.
2)	Assign binary variables in F_x to those in $\mathbb{Z}$. Denote the resulting convex polygonal region $\tilde{F}_x$.

Table 4. Function to form convex region from MILP solution.

Lemma 3 (Finite Time). *The algorithm described in Table 3 is guaranteed to terminate in bounded time.*

Proof: The polygonal nonconvex feasible state region F_x is defined by a disjunction of linear constraints. At each iteration of RobustNonconvexMain, a region defined by subsets of these linear constraints are removed from the search space. Since there are a countable number of linear constraints, there are a countable number of such subsets. If all constraint subsets are removed, the nonrobust problem becomes infeasible and $\hat{h}_i = \infty$, at which point the algorithm terminates. Hence termination is guaranteed in bounded time. □

V. Simulation Results

The algorithm RobustNonconvexMain was applied to a problem of chance constrained path planning for a UAV in a 2D environment with uncertain localization acting under wind disturbances. Following the approach of Ref. 6 we model the aircraft as a double integrator with velocity limits and turn rate constraints.

	Function PruneSearchSpace$(\tilde{\mathbb{U}}^r, \tilde{K}, \tilde{F}_x, F_x', \mathbb{Z})$ **returns** F_x'
1)	Determine which constraints defining the convex set $\tilde{F}_x$ are tight in the optimal solution $\{\tilde{\mathbb{U}}^r, \tilde{K}\}$ by finding those constraints with nonzero dual values.
2)	Determine the subset of the binary variables $\mathbb{Z}$ corresponding to these constraints. In other words, determine which variables in $\mathbb{Z}$ were set to zero to enforce the tight constraints. Denote the vector of these variables $\mathbb{Z}^{sub}$.
3)	Remove from F_x' the region for which every binary in $\mathbb{Z}^{sub}$ is zero. This is equivalent to adding to the MILP formulation of F_x' the constraint $sum(\mathbb{Z}^{sub}) \geq 1$.

Table 5. Function to remove region from nonconvex search space.

Denoting the position vector of the UAV as $[x\ y]'$, the system is defined as:

$$\mathbf{x} = \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \quad A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ \Delta t & 0 \\ 0 & 0 \\ 0 & \Delta t \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad B^w = B. \tag{17}$$

Initially the UAV is moving North with velocity $0.5m/s$. Localization uncertainty is modeled by Gaussian uncertainty in the initial state $\mathbf{x}_0$ while wind disturbances are modeled as Gaussian white process noise. The statistics of these random variables are given by:

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} \quad P_0 = \begin{bmatrix} 2.5 \times 10^{-3} & 0 & 0 & 0 \\ 0 & 2.5 \times 10^{-7} & 0 & 0 \\ 0 & 0 & 2.5 \times 10^{-3} & 0 \\ 0 & 0 & 0 & 2.5 \times 10^{-7} \end{bmatrix} \quad \bar{\mathbf{w}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad var(\mathbf{w}) = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \times 10^{-5}. \tag{18}$$

The maximum velocity of the UAV is $1.0m/s$ and the maximum acceleration is $0.25m/s^2$. These magnitude constraints were approximated using an 8-sided inscribing polygon as described by Ref. 7. A time horizon of 10 steps was used, with $\Delta t = 2s$. We constrain the system state to be in the goal region at the final time step, and to avoid all obstacles at all time steps. In the optimizations we minimize fuel use, defined as:

$$fuel = \sum_{k=0}^{N} \Big(|\ddot{x}_k| + |\ddot{y}_k| \Big). \tag{19}$$

We implemented the RobustNonconvexMain algorithm in Matlab using the YALMIP interface. The GLPK package was used to solve MILP problems, while SPDT3 was used to solve SOCP problems. The results shown here were generated on a 2.4GHz Macbook Pro with 4GB of RAM. Figure 4 shows the globally optimal robust path found by RobustNonconvexMain for two different obstacle maps with a maximum probability of failure of 0.01. One of these maps is easy for the RobustNonconvexMain algorithm, because the optimal robust solution is close to the optimal nonrobust solution. The other is hard, because the optimal nonrobust solution travels through a narrow corridor too risky for a robust solution. This means that RobustNonconvexMain has to spend more time searching for the optimal robust solution. We use these examples to illustrate the performance of the algorithm at two extremes.

Figure 5 shows the convergence of RobustNonconvexMain algorithm to the globally optimal solution. We show the cost of the best robust feasible solution found so far and the best lower bound on the robust feasible cost. The lower bound is provided by the cost of nonrobust feasible solutions. Global optimality is proven when the gap between the two values reaches zero. Figure 5 shows that the easy map yields a provably optimal solution in $2.9mins$ whereas the hard map requires $118mins$. Notice that in both cases the globally optimal solution is found in a small fraction of the total running time; the majority of time is spent *proving* global optimality. This suggests that, in practice, the algorithm can be terminated early without a substantial deterioration in the returned solution.

The true probability of failure for the globally optimal solutions in Figure 4 was estimated by performing 10^8 Monte-Carlo simulations. For both the specified maximum probability of failure was 0.01, however the true value was far below this; the easy map had a probability of failure of 1.0×10^{-6} and the hard map had a probability of failure of 5.0×10^{-7}. This indicates a high level of conservatism, which arises from the bounding approach used in the ConservativeConvexPlanning method of Ref. 15. Reducing this conservatism while guaranteeing chance constrained satisfaction is an open problem.

VI. Conclusion

In this paper we presented a novel method for optimal chance constrained path planning with feedback design. Unlike previous approaches to chance constrained path planning, the new approach optimizes the feedback gain as well as the reference trajectory. The new approach couples a fast, nonconvex solver that

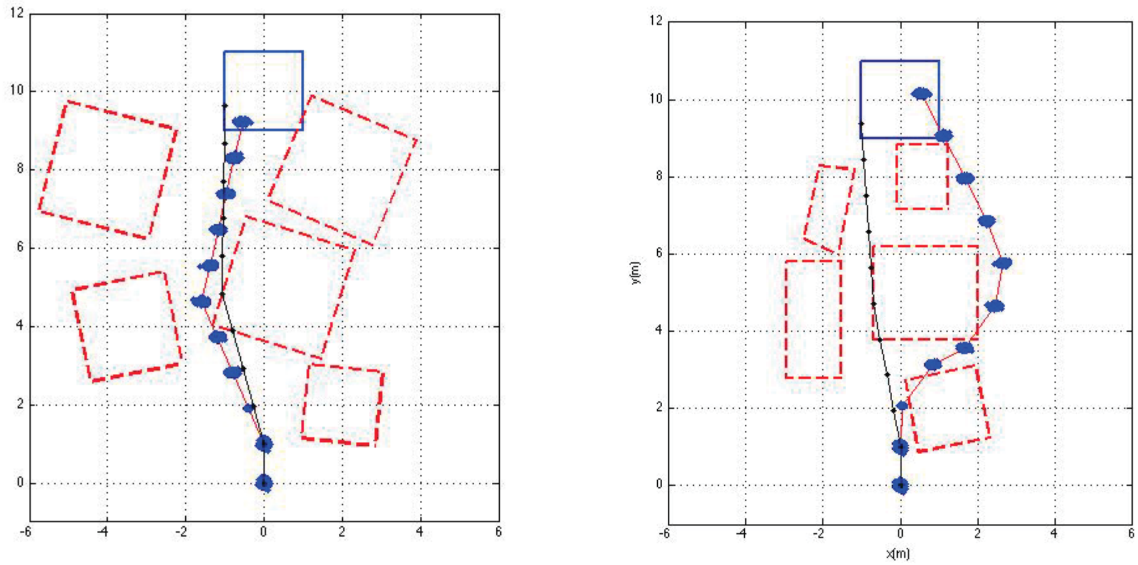

Figure 4. Globally optimal robust solutions for easy (left) and hard (right) maps. The reference trajectory is shown in red (thin) and the distribution of the position is represented using 1000 random samples. Also shown in black (thin) is the optimal nonrobust solution. In the easy map the optimal robust solution is close to the optimal nonrobust solution, so the algorithm terminates quickly. in the hard map the narrow corridor at $[-1, 5]$ means that the optimal robust solution is far from the optimal nonrobust solution. Nevertheless the algorithm eventually finds the optimal robust solution. In this example $\delta = 0.01$. Note that the state constraints are only imposed at the discretization points; obstacle 'jumpovers' can be avoided by standard tightening approaches.[18]

does not take into account uncertainty, with existing robust approaches that apply only to convex feasible regions. By alternating between robust and nonrobust solutions, the new algorithm guarantees convergence to a global optimum. We apply the new method to the problem of robust path planning for a UAV and show that the algorithm finds the globally optimal robust solution early, before performing additional computation to prove global optimality.

Appendix

Here we provide details of the Second Order Cone Program that is equivalent to the conservative convex planning problem (Def. 4). For details of the derivation, refer to Ref. 15. First, define:

$$Q = K(I - G_{yu}K)^{-1}. \tag{20}$$

Now define a polygonal feasible region for $\mathbb{X}$ as a conjunction of linear constraints:

$$\bigwedge_{j=1:N} \mathbf{a}_j' \mathbb{X} \geq b_j. \tag{21}$$

As shown by Ref. 15, the problem in Def. 4 is equivalent to the Second Order Cone Program given by:

$$\begin{aligned}
&\text{Minimize } h(\mathbb{U}^r, \mathbb{X}^r) \text{ over } \mathbb{U}^r \text{ and } Q \\
&\text{subject to} \\
&\mathbb{U}^r \in F_u \\
&Q \in \mathcal{K} \\
&\nu_j(Q) + \mathbf{a}_j' \mathbb{X}^r \leq b_j \quad \forall j \\
&\nu_j(Q) = r||\mathbf{a}_j'[(G_{xx} + G_{xu}QG_{yx})F_P) \quad (G_{xw} + G_{xu}QG_{yw})F_W]||_2 \quad \forall j \\
&\mathbb{X}^r = G_{xx}\bar{\mathbf{x}}_0 + G_{xu}\mathbb{U}^r + G_{xw}\bar{\mathbb{W}},
\end{aligned} \tag{22}$$

where $F_P = \sqrt{P_0}$ and $F_W = \sqrt{V}$.

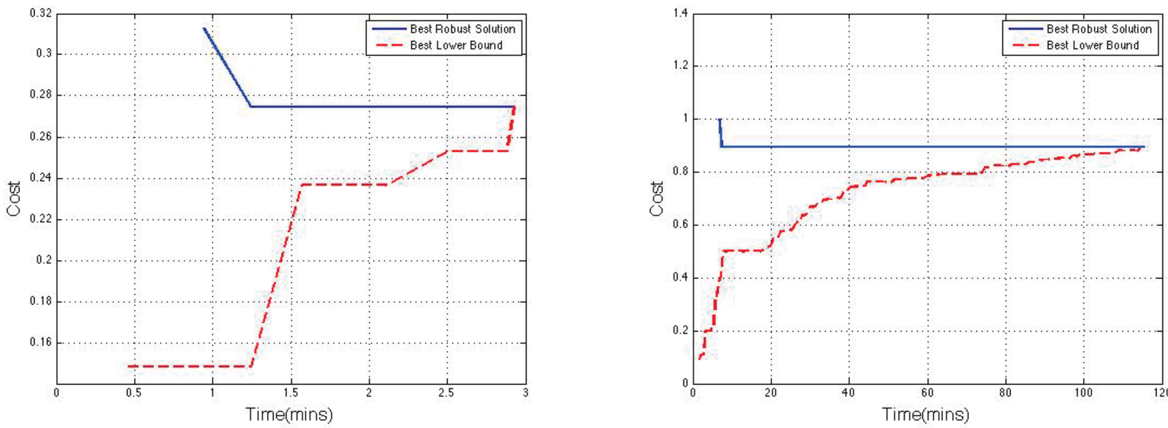

Figure 5. Convergence of RobustNonconvexMain algorithm to globally optimal solution for easy map (left) and hard map (right). Shown are the cost of the best robust feasible solution found and the best lower bound on the robust feasible solution. In both cases the globally optimal solution is found early in the process, with the majority of time being used to prove global optimality.

Acknowledgement

The research described in this paper was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Zhou, K. and Doyle, J. C., *Essentials of Robust Control*, Prentice Hall, 1997.

[2] Bemporad, A. and Morari, M., "Robust Model Predictive Control: A Survey," *In Robustness in Identification and Control*, edited by A. T. A. Garulli and A. Vechino, Springer-Verlag, 1999, pp. 207–226.

[3] Açıkmeşe, B. and Carson, J. M., "A Nonlinear Model Predictive Control Algorithm with Proven Robustness and Resolvability," *Proceedings of the American Control Conference*, June 2006, pp. 887–893.

[4] Carson, J. and Açıkmeşe, B., "A model predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006.

[5] Barr, N. M., Gangsaas, D., and Schaeffer, D. R., "Wind Models for Flight Simulator Certification of Landing and Approach Guidance and Control Systems," Paper FAA-RD-74-206, 1974.

[6] Schouwenaars, T., Moor, B. D., Feron, E., and How, J., "Mixed Integer Programming for Multi-Vehicle Path Planning," *Proc. European Control Conference*, 2001.

[7] Richards, A. and How, J., "Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming," *Proc. American Control Conference*, 2002.

[8] Blackmore, L., Li, H. X., and Williams, B. C., "A Probabilistic Approach to Optimal Robust Path Planning with Obstacles," *Proceedings of the American Control Conference*, 2006.

[9] Blackmore, L., "A Probabilistic Particle Control Approach to Optimal, Robust Predictive Control," *Proceedings of the AIAA Guidance, Navigation and Control Conference*, 2006.

[10] Blackmore, L., Bektassov, A., Ono, M., and Williams, B. C., "Robust, Optimal Predictive Control of Jump Markov Linear Systems using Particles," *Hybrid Systems: Computation and Control, HSCC*, edited by A. Bemporad, A. Bicchi, and G. Buttazzo, Vol. 4416 of *Lecture Notes in Computer Science*, Springer Verlag, 2007, pp. 104–117.

[11] van Hessem, D., Scherer, C. W., and Bosgra, O. H., "LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints," *Proceedings of the Control and Decision Conference*, 2001.

[12] van Hessem, D. and Bosgra, O. H., "Closed-loop stochastic dynamics process optimization under state and input constraints," *Proceedings of the Control and Decision Conference*, 2002.

[13] van Hessem, D. and Bosgra, O. H., "A full solution to the constrained stochastic closed-loop MPC problem via state and innovations feedback and its receding horizon implementation," *Proceedings of the Control and Decision Conference*, 2003.

[14] van Hessem, D. and Bosgra, O. H., "Closed-loop stochastic model predictive control in a receding horizon implementation on a continuous polymerization reactor example," *Proceedings of the Control and Decision Conference*, 2004.

[15] Hessem, D. H. V., *Stochastic Inequality Constrained Closed-loop Model Predictive Control*, Ph.D. thesis, Technische Universiteit Delft, Delft, The Netherlands, 2004.

[16] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[17] ILOG, "ILOG CPLEX User's guide," 1999.

[18] Kuwata, Y., *Trajectory Planning for Unmanned Vehicles using Robust Receding Horizon Control*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2007.